

EILCO-LOGO-2022.png

ÉCOLE D'INGÉNIEURS DU
LITTORAL-CÔTE-D'OPALE

TP : RÉSEAUX INDUSTRIELS ET SUPERVISION

Interface Pygame



Auteur: Fono Colince

Supervisé par:
Mr. PIERRE CHATELAIN

Date:
May 19, 2024

Contents

1	Introduction	2
2	interface personnelle	2
3	Pendant le cycle de remplissage	2
4	Pendant le cycle de vidange	2
5	Annexe	2
5.1	La liste des icônes créé	2
5.2	Un diagramme des classes	2
5.3	Vos codes sources	2

1 Introduction

Dans ce TP, vous allez utiliser un simulateur d'automate qui simulera un automatisme industriel. Vous allez ensuite communiquer avec cet automate en utilisant le protocole Modbus pour produire une supervision.

2 interface personnelle

3 Pendant le cycle de remplissage

4 Pendant le cycle de vidange

5 Annexe

5.1 La liste des icônes créé

5.2 Un diagramme des classes

5.3 Vos codes sources

```
1 import pygame
2 import time
3 from math import pi, cos, sin
4 import random
5
6 # Initialisation de Pygame et de l'cran
7 pygame.init()
8 ecran = pygame.display.set_mode((800, 600))
9 pygame.display.set_caption("Simulation de R réservoir de Liquide")
10 clock = pygame.time.Clock()
11
12 # Couleurs
13 GREEN = (0, 255, 0)
14 BLACK = (0, 0, 0)
15 RED = (255, 0, 0)
16 BLUE = (0, 0, 255)
17 GREY = (169, 169, 169)
18
19 # Fonction pour dessiner le r réservoir
20 def draw_tank(surface, x, y, width, height, fill_level):
21     pygame.draw.rect(surface, BLACK, (x, y, width, height), 2)
22     # Dessin du liquide
23     liquid_height = fill_level / 100 * height
24     pygame.draw.rect(surface, BLUE, (x + 2, y + height -
25         liquid_height, width - 4, liquid_height))
26     # Ajout d'un effet de d grad pour le liquide
27     for i in range(1, int(liquid_height), 2):
28         alpha = 255 - int(255 * (i / liquid_height))
29         s = pygame.Surface((width - 4, 2), pygame.SRCALPHA)
30         s.fill((0, 0, 255, alpha))
```

```

30         surface.blit(s, (x + 2, y + height - i))
31
32 # Fonction pour dessiner un indicateur de niveau de liquide
33 def draw_level_indicator(surface, x, y, fill_level):
34     font = pygame.font.Font(None, 36)
35     level_text = font.render(f"{fill_level:.1f}%", True, BLACK)
36     surface.blit(level_text, (x, y))
37
38 # Fonction pour dessiner des cercles (bubbles) dans le liquide
39 def draw_bubbles(surface, x, y, width, height, fill_level,
40                 bubbles):
41     liquid_height = fill_level / 100 * height
42     for bubble in bubbles:
43         bubble_y = y + height - bubble[1] * liquid_height / 100
44         pygame.draw.circle(surface, GREY, (x + bubble[0] * width
45         / 100, int(bubble_y)), bubble[2])
46
47 # Génération initiale des bulles
48 def generate_bubbles(num_bubbles):
49     return [(random.randint(10, 90), random.randint(10, 90),
50             random.randint(2, 5)) for _ in range(num_bubbles)]
51
52 # Animation des bulles
53 def animate_bubbles(bubbles, height):
54     for i in range(len(bubbles)):
55         bubbles[i] = (bubbles[i][0], (bubbles[i][1] - 0.5) % 100,
56                       bubbles[i][2])
57
58 # Initialisation des bulles
59 bubbles = generate_bubbles(15)
60
61 def main():
62     continuer = True
63     fill_level = 50.0 # Niveau initial de liquide 50%
64
65     while continuer:
66         for event in pygame.event.get():
67             if event.type == pygame.QUIT:
68                 continuer = False
69             elif event.type == pygame.KEYDOWN:
70                 if event.key == pygame.K_q:
71                     continuer = False
72                 elif event.key == pygame.K_UP:
73                     fill_level = min(100.0, fill_level + 1.0)
74                 elif event.key == pygame.K_DOWN:
75                     fill_level = max(0.0, fill_level - 1.0)
76
77         ecran.fill((224, 224, 224)) # Couleur de fond
78
79         fill_level = fill_level+0.01
80
81         # Dessiner le r servoir avec le niveau de liquide
82         draw_tank(ecran, 100, 100, 100, 100, 100)
83         draw_level_indicator(ecran, 100, 200, 100)
84
85         draw_tank(ecran, 600, 100, 100, 100, 100)
86         draw_level_indicator(ecran, 600, 200, 100)

```

```

83
84     draw_tank(ecran, 350, 400, 100, 100, fill_level)
85     draw_level_indicator(ecran, 350, 500, fill_level)
86
87     # Dessiner et animer les bulles
88     draw_bubbles(ecran, 350, 400, 100, 100, fill_level,
89     bubbles)
90     animate_bubbles(bubbles, 300)
91
92     pygame.display.flip()
93     clock.tick(30)
94
95     pygame.quit()
96
97 if __name__ == "__main__":
98     main()

```

Code source 1: .